

Linux Pre-Install and Install

Advanced Track: Lecture 3

Edric Xiang

(slides partially adapted from Christopher Cooper)

This week we'll be installing...



Why Arch?

m i n i m a l: Relatively little included by default, user configures what they want.

Rolling-release: Install once, frequently updated afterward.

Arch User Repository (AUR): Repository of community-maintained packages for Arch.

Arch Wiki: Probably the most comprehensive documentation of any distribution.

Learning opportunity: More involved installation process deepens your understanding!

... ok, then why NOT Arch?

Stability*: Things may break. Are you willing to spend time to learn how to tinker and fix problems that may arise?

Difficult installation: No GUI installer; installation is done 100% through the shell.

* You get newer versions of packages faster, which is a double edged sword: you may experience bugs/instability, but you also get bug fixes earlier.

What makes Arch different?

Let's compare it to another popular Linux distro:

ubuntu 



Arch vs Ubuntu



- Command-line installation
- No desktop environment/window manager included
- Comes with the bare minimum to have a working OS at the end of installation
- Rolling updates (approx. monthly)
- Official packages + user maintained packages through the AUR

- User-friendly GUI installation
- Comes with GNOME 3 desktop environment
- Bundles some extra things like proprietary graphics card drivers
- New release twice a year
- Official packages

Installation Concepts

1. Booting the Installation Media

2. Initial Setup
3. Partitioning
4. Setting up Filesystems
5. Encrypting your Filesystem
6. Mounting your Filesystems
7. Installation (pacstrap)
8. Configuring the System

Booting the Installation Media

- Digital Ocean virtual machines run Ubuntu 18.04
- Arch will be installed on a VM... within the Ubuntu VM
- Download Arch installation media from our own OCF hosted mirror
- Conceptually the same as inserting a Arch live USB to install
- Not a super important step, so just follow the detailed instructions in the lab!

Installation Concepts

1. Booting the Installation Media

2. Initial Setup

3. Partitioning

4. Setting up Filesystems

5. Encrypting your Filesystem

6. Mounting your Filesystems

7. Installation (pacstrap)

8. Configuring the System

Initial Setup

- These should be fine by default, [but if you want to be thorough and check...](#)
 - Verify you booted in UEFI mode (the command we give you does this)
 - Set the keyboard layout (defaults to normal US layout)
 - Connect to the internet (VM should automatically do this)
 - Update the system clock (should automatically sync if you're connected to the internet)

Installation Concepts

1. Booting the Installation Media
2. Initial Setup

3. Partitioning

4. Setting up Filesystems
5. Encrypting your Filesystem
6. Mounting your Filesystems
7. Installation (pacstrap)
8. Configuring the System

Partitioning: What is it?

Dividing a storage device into different sections:

/dev/sda		
/dev/sda1	/dev/sda2	/dev/sda3
/boot or /efi	/	[swap]
EFI System Partition	Linux x86-64 root (/)	Linux swap
512 MB	4 GB	512 MB

Partitioning: More technical details

- Fixed-size subset of disk treated by OS as a single unit
- This happens below the filesystem level:
 - **G**UID **P**artition **T**able (**GPT**)
 - **M**aster **B**oot **R**ecord (**MBR**)
- Disks need to be partitioned before you can install filesystem on a partition
- This is necessary because the partition map tells the bootloader where to find the boot/root partitions/filesystems
- For the lab, you will be using `fdisk` to create partitions

Partitioning: Naming conventions

- Linux “everything is a file” philosophy: file interfaces for storage devices and partitions live in the `/dev` directory
 - Storage devices (e.g. hard disks/SSDs): `/dev/sda`, `/dev/sdb`, `/dev/sdc`, etc.
 - Partitions on `/dev/sda`: `/dev/sda1`, `/dev/sda2`, `/dev/sda3`, etc.
- Warning: `/dev/sdXX` is not persistent! `/dev/sda` can become `/dev/sdb`
- Refer to drives/partitions by UUID, which will never change

Partitioning: Partitioning schemes

- Some people like making separate partitions for certain directories:
 - **/home**: Contains user files, and typically folders like Documents, Pictures, Downloads, etc. Making a separate /home partition can let you reinstall/switch distros without losing your personal files.
 - **/var**: Contains logs, which can cause your disk space to be eaten up by massive log files in rare cases. Making a separate /var partition allocates it a fixed amount of space so that it can't overrun your main partition.
- Swap partition sizing:
 - Lots of different recommendations; traditionally swap size == RAM size
 - For a modern PC (>8GB of RAM) you can get away with smaller (or even no swap)
 - ... or ditch a swap partition entirely and use a swap file

Installation Concepts

1. Booting the Installation Media
2. Initial Setup
3. Partitioning
- 4. Setting up Filesystems**
5. Encrypting your Filesystem
6. Mounting your Filesystems
7. Installation (pacstrap)
8. Configuring the System

Filesystems

- We've partitioned our storage, but those partitions are currently blank
- How to store data on those partitions in an organized manner? A filesystem
- Many filesystems with different features and compatibility:
 - FAT32: old and simple FS still found on flash drives/SD cards; max 2GB file size
 - NTFS: Microsoft's proprietary FS for Windows
 - APFS: Apple's proprietary FS, optimized for SSDs and encryption
 - **ext4: most common Linux FS, succeeding ext3 in 2008**

Filesystems: Formatting

- We must **format** our partitions with an appropriate file system
- EFI system partition (ESP): format with **FAT32**
 - `mkfs.fat -F32 /dev/[your ESP here]`
 - FAT filesystem recommended for compatibility with other OSes (e.g. dual-booting Windows)
 - UEFI specification (officially) only supports FAT filesystems (FAT12, FAT16, FAT32)
- root: format with **ext4**
 - ... but before that, we'll be setting up encryption first!
- swap: not really a filesystem, but still needs to be set up
 - `mkswap /dev/[your swap partition here]`
 - `swapon /dev/[your swap partition here]`

Installation Concepts

1. Booting the Installation Media
2. Initial Setup
3. Partitioning
4. Setting up Filesystems
- 5. Encrypting your Filesystem**
6. Mounting your Filesystems
7. Installation (pacstrap)
8. Configuring the System

Disk Encryption

- Is the data on your disk safe if you have an account password? NO
- Someone could plug your disk into their machine and read everything on it
- Solution: encrypt the disk
 - Data looks random (if you properly wipe it first by writing random data)
 - Relatively low overhead (takes a little extra time to decrypt when booting)

Installation Concepts

1. Booting the Installation Media
2. Initial Setup
3. Partitioning
4. Setting up Filesystems
5. Encrypting your Filesystem

6. Mounting your Filesystems

7. Installation (pacstrap)
8. Configuring the System

Mounting

- Windows: each partition/device is assigned a drive letter
 - Example: C:, D:, E:, F:, etc.
- Unix: Other partitions/devices must be **mounted** in the directory tree of your root filesystem to be able to access them
 - Example: plug in a USB drive → **mount** the USB drive at /mnt/myusbdrive → USB drive's contents appear in the directory /mnt/myusbdrive
 - Often done automatically by OS, but we need to do it manually during Arch install
- We need to mount our /boot and root / filesystems so that we can view/access them from our Arch live installation environment
 - After we mount /boot and / (which are currently blank), we can install the relevant files to those partitions (bootloader, Linux kernel, etc.)

Installation Concepts

1. Booting the Installation Media
2. Initial Setup
3. Partitioning
4. Setting up Filesystems
5. Encrypting your Filesystem
6. Mounting your Filesystems
- 7. Installation (pacstrap)**
8. Configuring the System

Installation: Basic packages needed for Arch

What does the pacstrap command do?

1. Underneath, installs a bunch of packages (including the Linux kernel) needed for your system to work
2. Installs a
 - a. kernel (/boot/vmlinuz*)
 - b. initrd (/boot/initrd.img*): “initial ramdisk”, used as part of the Linux startup process
 - c. init system (systemd): “initialization system”, handles mounting filesystem after startup, starting services and daemons, etc.
 - d. package manager (pacman)
 - e. minimal set of useful packages (“[base](#)”, etc.)

Installation Concepts

1. Booting the Installation Media
2. Initial Setup
3. Partitioning
4. Setting up Filesystems
5. Encrypting your Filesystem
6. Mounting your Filesystems
7. Installation (pacstrap)
- 8. Configuring the System**

Configuration

- Let's cover some of the more important configuration steps...
- Generate `/etc/fstab` using `genfstab`: “filesystem table”
 - Defines how partitions should be mounted to the filesystem
 - Used by `systemd` to mount filesystems in their proper place during boot
- `chroot`: “change root”
 - Until this point, you've been operating in the context of the Arch live environment
 - Now that our base Arch OS has been installed to our disk, we need to change our root directory to be our installed filesystem (`/mnt` on live environment → root (`/`) in installed environment)
- Create `/etc/hostname`: network configuration
 - Basically just creating a mapping between `127.0.0.1` and `localhost`

Configuration cont.

- Creating new initramfs with modified mkinitcpio config
 - initramfs: “initial RAM filesystem”, small pre-init environment for loading/setting stuff up
 - mkinitcpio: a bash script used to create initramfs
- Because we have an encrypted disk, we need to modify `/etc/mkinitcpio.conf`
 - Add two hooks: `keymap` and `encrypt`
 - Move `keyboard` hook before `keymap` (order matters!)
- You need to have access to the keyboard/keymap and encrypt hooks so that you can type your password to decrypt your filesystem

I've finished installation, what now?

- “Wait, that’s it?” Unfortunately, the end of this lab is pretty anticlimactic...
 - Arch doesn’t come bundled with a desktop environment, so all you’ll see at the end is a terminal prompt blinking back at you
- Next week’s lecture/lab will cover post-installation setup:
 - Making users and groups
 - Setting up a firewall/securing your network
 - Installing basic CLI utilities
 - etc.
- If you want to use Arch with a desktop environment:
 - Install Arch on a VM using VirtualBox on your personal computer (see Youtube tutorials)
 - During the pacstrap step, include “gnome” or your preferred desktop environment or window manager (or just install with pacman at the end)

Notes on this week's lab

- Installing Arch can be tough! Give yourself at least a few hours, don't try to do it all last minute.
- Use the Arch Wiki! We tried to make the lab instructions very detailed, but just in case:
 - [Installation Guide](#): rather minimal, but click on the links within the article for more detailed information about any step!
 - Don't be afraid to Google stuff or ask a question on Piazza!

How does booting actually work?

- MB signals PSU to supply power, which triggers MB to start CPU
- CPU resets, performs **Power-On Self Test**, loads BIOS or UEFI into memory
 - BIOS selects boot method, e.g. block storage, network
 - Block: BIOS reads first sectors of disk which contain MBR/GPT metadata, executes data from here to load into 2nd-stage bootloader
 - Network: BIOS has NIC gain network access (DHCP) and requests netboot, downloads/unpacks kernel/initrd directly into memory, uses this as 2nd-stage bootloader
 - UEFI is configured in NVRAM to look for an EFI System Partition
 - If no configured partition, look at everything that is FAT32 and look for certain files
- 2nd-stage bootloader (e.g. GRUB/systemd-boot) takes over, reads config for OS location, loads kernel into memory and jumps to kernel init
 - loads initramfs (initial RAM filesystem) or tells kernel where to find it

Booting the Installation Media

Some Initial Stuff

Partitioning

Setting up Filesystems

Encrypting your Filesystem

